ROUTING AND TRANSMITTAL SLIP Date 2 JUN 1981

TO: (Name, office symbol, room number, building, Agency/Post)	Initials	Date
1. AEO/DDA	T	2 JUN 1981
2.		
3.		
4. D/ODP		
5. cc C/IHSA		

Action	File	Note and Return
Approval	For Clearance	Per Conversation
As Requested	For Correction	Prepare Reply
Circulate	For Your Information	See Me
Comment	Investigate	Signature
Coordination	Justify	

REMARKS

4-FYI
5-FYI

DO NOT use this form as a RECORD of approvals, concurrences, disposals, clearances, and similar actions

FROM: (Name, org. symbol, Agency/Post)	Room No.—Bldg.
AEO/DDA	Phone No.

5041-102 OPTIONAL FORM 41 (Rev. 7-76)
Prescribed by GSA
FPMR (41 CFR) 101-11.206

☆ GPO : 1980 O - 311-156 (17)

81-1159

STAT

AEO/DDA: (2 Jun 81)

Distribution:
Orig YRS - D/ODP w/atts
1 - IHSA w/atts
1 - DDA Subject w/atts

STAT

HOS

HIGHER
ORDER
SOFTWARE, INC.

806 MASSACHUSETTS AVENUE
CAMBRIDGE, MA. 02139
TELEPHONE 617 • 661 • 8900

22 May 1981

William J. Casey
Director, Central Intelligency Agency
Washington, DC 20505

Dear Mr. Casey:

The C^{3}I environment continues to expand. Your comments in the May/June issue of Signal magazine reflect your concern and the need for continued work in this area. In that same issue the article "C^{3}I Covenants: An Architectural Tool for the 80's" presented an approach for the "system architect" of C^{3}I systems. We currently market an automated tool "USE.IT" that provides the capability to support that systems architect.

Enclosed is information on the tool, and an announcement of a workshop on the methodology behind that tool. I think this information will be of interest to you and the people in your organization.

Sincerely,

Margaret H. Hamilton
President

MHH/gl

Enclosures

HOS 1800

USE.IT: OVERVIEW AND PROSPECTUS

USE.IT (User System Evaluation and Integration Tool) is an integrated family of tools developed by Higher Order Software (HOS), Inc. for automating a system's life cycle. USE.IT can be considered as a "development machine" in that processes in a life cycle, traditionally thought of as manual ones, can now be automated. In fact, the definition process, itself, is supported by automation with USE.IT. Whereas the conventional life cycle approach is historical in nature in that features have been added on an ad hoc basis as needed, the HOS life cycle approach is functional in nature.

USE.IT is based on the HOS methodology. HOS, a systems theory based on analysis of large complex systems development, concerns, among other aspects, the definition of systems so as to eliminate data and timing conflicts. It is applicable to all phases and all aspects of system development. A number of systems using these techniques have been developed and implemented with considerable success.

There are many unique aspects about HOS that should be of great interest to system developers. For example, the very use of HOS can eliminate a large class of errors from your system development process. HOS also identifies concurrencies inherent in your systems, allowing efficient use of multiprocessing resources. One of the more exciting properties to result from specifying a system with HOS is that it offers an opportunity to build a whole new breed of tools not heretofore thought possible. An example of just such a collection of tools is USE.IT.

USE.IT has several components which make up its integrated whole. One component is the specification language, AXES. AXES has been developed to provide a representation of systems in terms of the 3 basic HOS units: data types, functions, and control structures. AXES is a specification language for defining mechanisms which are used to define systems. This means that AXES can be used as a universal semantic basis for the representation of any system; yet, AXES allows varying dialects defined in terms of this common set of semantics. With this capability, AXES provides a means for users and developers to speak the same language. Use of common mechanisms is obtained by either common use of the same mechanisms in an AXES library or by various derivations from existing mechanisms in an AXES library.

Another component, the Analyzer, provides static verification of AXES specifications. This process ensures consistent system definitions. It also assists in the interactive building of AXES specifications.

A given AXES specification can be itself treated as data in another system and transformed to another representation--into a higher order language (HOL) or machine code, for example. Different representations generated in this manner are referred to as layers of implementation in HOS. The process of transforming a specification for implementation in a given machine or language may be thought of as a resource allocation process. Because the structure is completely specified, the resource allocation can be handled by another component, the automated resource allocation tool (RAT) in the USE.IT system. While translation of an AXES specification into an HOL description is possible, it is also possible for a RAT to perform the resource allocation process directly into executable code for a particular machine environment. In this way the AXES specification can, itself, be the last point at which human intervention is required.

With a USER friendly component of USE.IT, AXES specifications can be transformed to other representations by translation of a proper subset of its properties. Automated means can provide representation of an AXES specification in data flow structures, priority structures, structured design diagrams, SADT charts, PSL/PSA, IDEF, Petri nets, SREM and other representations. Thus, users familiar with one of these representations may have a "friendly" interface to HOS systems semantics. With USE.IT, a user may also start with his own selected syntax. The decision support capability would then help him to make it behave like HOS. Thus, for example, one user may want to depict his system definition in English, another in a "front-end" syntax or, still another in a program design language, or even an HOL, for that matter. In each case, USE.IT has the ability, in its user friendly package, to translate from a particular user's environment to the development environment of USE.IT. Thus these users who are already committed to other syntaxes can be brought to a common meeting ground in order that they, too, can capitalize on the benefits of the USE.IT automated development process. The user friendly package is reconfigurable depending on each particular user's need.

In its final form, USE.IT presents a radical departure from the traditional approach. It not only provides a strict separation of the specification of a system from its implementation, but it also permits a totally automated implementation of a system from a completely machine-independent specification. A friendly interface with automated decison support is provided by the user friendly package; the system is specified with AXES; the definition produced is checked for consistency by an analyzer; and the verified definition is resource allocated by a RAT developed for the particular target machine environment or if so desired to an intermediate environment such as ADA, FORTRAN or COBOL.

Once the RAT is fully implemented in USE.IT, it will provide a drastic reduction in the requirement for programming personnel in system development activities. That is, since AXES specifications can be translated directly into executable code, actual programming requirements can be entirely eliminated for most applications.

USE.IT will provide your computer facility with these major capabilities:

1. a requirements definition language with edit and display of HOS system models in a variety of graphics or text-oriented syntaxes. (Tektronix compatible graphics.)

2. An analyzer which checks system definition modules for syntax errors and for interface errors between modules to ensure consistency of system definition.

3. A manager for a library of system modules, allowing HOS defined modules to act as an evolving custom design language in your environment.

4. A resource allocator to automatically generate efficient higher-level-language code from HOS specifications.

5. A user friendly module which provides decision support for a user in defining his requirements.

Each user is able to choose a selected configuration of USE.IT for his own needs. An example of one of these configurations of USE.IT is FAME which is used as a front-end definition package. FAME has a user friendly module which uses AXES syntax directly and an analyzer module to ensure that the front end definitions are consistent and logically complete.

The USE.IT tools are themselves "coded" as HOS specifications, allowing easy re-targeting to any machine environment. As USE.IT is not restricted to any particular environment or applications area, it is the most powerful system development aid currently available!

PRODUCT SCHEDULE

FAME, one packaged configuration of USE.IT, is currently available off-the-shelf for the VAX/VMS environment, incorporating capabilities 1, 2 and 3 ($22,500 including user group). The full-blown multi-environment USE.IT incorporating capabilities 1, 2, 3 and 4 is scheduled for release September 1981.

5 is available upon request for a particular family of users.